# Chapter 1
# Internet-based Ground Operations for Mars Lander and Rover Missions

*Paul G. Backes*, Kam S. Tso†, Jeffrey S. Norris*, Gregory K. Tharp†*

**Jet Propulsion Laboratory, California Institute of Technology*

*†IA Tech Inc.*

## Abstract

The Web Interface for Telescience (WITS) has been developed to provide Internet-based ground operations for planetary lander and rover missions. This paper describes the design of WITS and its use in the Mars Polar Lander mission and for commanding the Field Integration, Development, and Operations (FIDO) prototype mars rover.

## 1.1 Introduction

The Web Interface for Telescience provides Internet-based ground operations for planetary lander and rover missions. WITS provides an integrated operations environment for operations at the central operations location, collaboration by geographically distributed scientists, and public outreach. Collaboration by geographically distributed scientists at their home institutions enables participation in missions by a greater number of scientists and reduces operations costs. Providing the same operations tool, with real mission data, to the public enables the public to have a better understanding of the mission and a more engaging mission experience. An initial version of WITS provided some of the features needed for a planetary rover mission [1]. Based upon experience with WITS as an evaluation and public outreach tool in the 1997 Mars Pathfinder mission [2] and rover field tests [3,4], WITS was redesigned and re-implemented to provide the features needed for planetary mission operations. This paper describes the new WITS system.

Through the use of WITS, the Mars Polar Lander (MPL) mission was the first planetary mission to utilize Internet-based ground operations to enable geographically distributed scientists to collaborate in daily mission command sequence generation. The same WITS system was used for the MPL mission and for commanding the FIDO rover. Other examples of Internet-based robot operation can be found in [5,6].

### 1.1.1 Mars Polar Lander Mission

The Mars Polar Lander landed near the south pole of mars on December 3, 1999 and was to perform an approximately three month mission [7]. Unfortunately, communication with the lander was never achieved so commanding the lander was not possible. Results shown in this paper are from commanding the lander in the University of California at Los Angeles (UCLA) MPL testbed. An artist's drawing of the lander is shown in Figure 1.1. The lander carried the Mars Volatiles and Climate Surveyor (MVACS) instrument suite. The mission operations center was at UCLA.

WITS served multiple purposes for the MPL mission. It was the primary operations tool for visualization of downlink data and generation of command sequences for the robotic arm (RA) and robotic arm camera (RAC). It was also used as a secondary tool for command sequence generation for the Stereo Surface Imager (SSI) stereo camera which was mounted on a lander mast, e.g., for visualizing footprints on the surface where SSI images would be taken. WITS also enabled Internet-based users to generate command sequences for the RA, RAC, and SSI. For example, scientists at the University of Arizona, who were responsible for the RAC and SSI, were able to generate sequence inputs from Arizona so that they would not have to be at the UCLA operations center for the whole mission. Also, a separate WITS system was provided to the general public to download to their home computers to enable them to plan and simulate their own missions.

### 1.1.2 FIDO Rover Operations

WITS was used to command the Field Integration, Development, and Operations (FIDO) rover in the two week field test at Silver Lake in the California Mojave desert in April 1999. The FIDO rover has been developed as a prototype vehicle for testing new planetary rover technologies and testing operations scenarios for the Mars'03 Athena rover mission [8,9]. The FIDO rover is shown in Figure 1.2. WITS supported operations in the field as well as remote operations over the Internet.

## 1.2 System Description

The WITS architecture is shown in Figure 1.3. The WITS database holds downlink data products and uplink sequence information. The WITS server provides communication between the WITS database and the WITS clients. The WITS clients are distributed over the Internet and provide the interface to the users to view downlink data and generate command sequences. Internet security was integrated between the server and clients. Communication between the client and server is implemented using Remote Method Invocation (RMI). The WITS database is a structured file system. Other tools, e.g., planetary ephemeris tools or other sequence tools, can interact with WITS by reading and writing to the WITS database or by direct communication with the WITS server.

The WITS client and server are implemented using the Java2 and Java3D programming languages. The client is run either as a Java applet using the appletviewer or as a Java application. Users must first download the Java Runtime Environment and Java3D.

### 1.2.1 Internet Security

A critical element in Internet-based mission operations is Internet security. Each day, a large amount of data is received from the spacecraft at the mission operations center and placed into a local database for processing and viewing by mission scientists. To enable collaboration in daily sequence generation by distant, Internet-based, scientists, a secure and efficient way to deliver the data to the remote scientists, as well as receive inputs from them, is needed. WEDDS, the WITS Encrypted Data Delivery System, was created to provide the required secure Internet-based communication for WITS. WEDDS was integrated with WITS for the MPL mission, but is designed to work with any mission application with little modification. WEDDS operates in a fashion that is transparent to the remote user. Files simply appear on the remote user's machine as they become available, and connections are made securely without any additional effort on the part of the user.

All WEDDS connections are authenticated using the NASA Public Key Infrastructure (PKI) [10]. After authentication, all WEDDS communications are made through SSL (Secure Sockets Layer) sockets and are encrypted using the Triple-DES-EDE3 algorithm [11,12]. The following are some of the advantages of WEDDS for Internet-based data distribution. 1) WEDDS does not require the remote users to request a data update. The data is automatically delivered as it becomes available. 2) The WEDDS administrator can specify on a user by user basis exactly which users will receive a particular type of file or directory. 3) Since WEDDS is written entirely in Java, it can run on most computers without modification. 4) WEDDS provides a high level of data security by using the SSL algorithm to perform authentication and the Triple-DES-EDE3 algorithm for encryption. The SSL protocol protects nearly every online commerce transaction, and breaking the Triple-DES-EDE3 encryption algorithm, even if every computer in the world was used, would take millions of years [13]. 5) WEDDS clients can be allowed to transfer files back to the mission control center. Files from users are stored on the server in a compressed, enveloped form that allows them to be scanned for hostile code. Since every client is authenticated using the NASA PKI, unauthorized users can not transmit files to the server.

WEDDS is implemented as two Java programs, a server and a client, using the publicly available Entrust Java Toolkit for its low level authentication and encryption tasks [14]. For each mission, there is typically one server, operating behind a mission firewall, and many clients, one on each remote user's machine.

Figure 1.4 illustrates the steps necessary for a single WEDDS transaction. Steps 1 and 2 occur once, before the beginning of the mission, while steps 3 through 9 occur for every transaction. In step 1, a remote user must obtain a security profile from the NASA Public Key Infrastructure (PKI), which requires appearing in person at a NASA center security office. A security profile is a set of files on a floppy disk, protected by a password, that contain a user's private key. A user needs his private key to positively identify himself online. The WITS server is also issued a security profile so that it can prove its identity to remote users. In step 2, each user must contact the mission administrator and request that their profile be given access to mission data.

Steps 3 through 9 are repeated for every transmission from the client to the server or from the server to the client. In steps 4 through 7, the server and client exchange digital "signatures", generated from their security profile. They verify these signatures by communicating with the NASA PKI. This process, SSL authentication, relies on the fact that it is nearly impossible for someone to generate another user's digital signature without that user's private key and password. In step 8, the last step in establishing an SSL connection, the client and server use the Diffie-Hellman key agreement protocol [15] to generate a unique symmetric encryption key that will be used to encrypt the traffic for this transaction. This encryption key is never sent as clear-text over the internet, and all subsequent traffic for this transaction is encrypted, making WEDDS transactions invulnerable to eavesdropping or "packet-sniffing" attacks. In addition, every WEDDS transaction is protected by a new encryption key.

### 1.2.2 Distributed Collaboration

Internet-based collaboration is achieved in WITS by providing Internet-based users with daily downlink data and allowing them to specify targets and generate command sequences and save them to the common server. The Internet-based users can see each others targets and sequences and can use target inputs from each-other's sequences. The Sequence window File pull-down menu enables sequences to be loaded from the common server (e.g., located at UCLA), or to be saved to the common server. Users can also save and load sequences to their local computers.

### 1.2.3 Downlink Data Visualization

Downlink data from the lander or rover is provided to the user via various views (Figure 1.5). Example views from the MPL mission are used here. The same views, but with rover data are used for FIDO rover commanding. Two windows provide the user with available data to be visualized. The Results Tree window (not shown in the figure) displays the available downlink data for the mission by date of downlink. The Plan window (Figure 1.5) displays available views for a specific plan. Each of these specific views has a specific set of downlink data it uses, so definitions of these views may be updated each day and put in the new plan. The user opens a view to visualize downlink data by clicking on the item. The various types of views are described below.

The Descent view (not shown in the figure) provides images taken from the spacecraft during descent to the surface and shows the landing location. For rover missions, the rover traverse path through the mission is shown. The Overhead view (Figure 1.5) shows the area around the lander or rover from above. Targets are displayed in the Overhead view, as well as selected points and view objects. Clicking on a point in the Overhead view causes the x,y position to be displayed at the clicked location. Clicking and dragging causes a ruler to be shown with the start and end points and distance displayed.

The Panorama view (Figure 1.5) is a mosaic of images taken by a stereo camera (on a mast on the lander in the MPL mission and on a deployable mast on the FIDO rover). Selecting a point in an image causes the x,y,z position on the surface to be displayed. The point can be turned into a known science target via a menu option. The Panorama view can be shown in 1/4, 1/2, and full scale. The view can also be shown in anaglyph stereo via a menu option. Clicking and dragging causes a ruler to be displayed with the start and end points x,y,z values and the distance and azimuth between the points. The Wedge view displays one image with various viewing options.

The Contrast Adjuster view (opened from a Wedge view pull-down menu) enables the contrast to be adjusted for a Wedge view image. The minimum and maximum desired pixel intensities are selected via scroll bars and then the pixel intensity values of the image are linearly stretched to have the selected pixel intensities become minimum (0) and maximum (255). The histogram of the initial and adjusted images are also shown.

The 3D view, shown in Figure 1.6, provides a 3D solid model visualization. Sequence simulation and state is visualized in the 3D view.

### 1.2.4 Sequence Generation

The views discussed above provide a means for visualizing downlink mission data. WITS also provides various windows and features for command sequence generation. WITS enables 3D locations to be turned into known targets to be used as parameters in commands. Targets are displayed in the various views as pink circles.

The Sequence window, shown in Figure 1.5, is used to generate a command sequence. A command sequence has a hierarchy of elements. The hierarchy, in descending order, is: Sequence, Waypoint, Request, Macro, Step. A request represents a high-level task. A macro is the functional element in WITS by which the user specifies commands and parameters. Macros have expansions into steps. A step is a low-level command that will be uplinked to the spacecraft. WITS can generate various format output sequences. The Sequence window shows the sequences in one plan. Multiple sequences can be displayed. A plan generally represents the planning elements to generate one command sequence to be uplinked to the lander or rover. The sequences are shown on the right hand side of the Sequence window. Supporting multiple sequences is useful for integration of subsequences from different scientists or subsequences for different instruments into the final uplink sequence. There are various sequence editing features in the Sequence window.

A list of macros which can be inserted into a sequence is shown on the left side of the Sequence window. Multiple lists of macros are available; choosing between macro lists is done via the pull-down menu above the macro list. A macro is inserted into a sequence by selecting the location in the sequence for it to be inserted and then double clicking on the macro in the macro list. Double clicking on a macro in the sequence causes the Macro window to pop up (Figure 1.7), where parameters for that macro are specified. A macro can generate view objects which are displayed in the views to indicate what the macro is producing. Figure 1.5 shows square outlines which are view objects for SSI imaging commands. They represent where images will be taken by the SSI in the current sequence. Above the dump pile is a view object of a RAC image and above the trench is a view object of a three-image RAC mosaic. View objects can also be generated by parsing the steps directly.

Resource analysis and rules checking are provided for sequences. The duration, energy, and data volume for each step of the sequence are computed and stored along with the cumulative duration, energy and data volume at each step. Sequences are also automatically checked to verify that they satisfy specified sequence rules.

The sequence state at a step is displayed when a user clicks on the step in the Sequence window. Step expansion and resource information is displayed at the top of the Sequence window and the system state is visualized in the 3D View.

## 1.3 Performance

### 1.3.1 Mars Polar Lander Mission Results

WITS was a part of the complete MPL mission ground operations system. A simplified diagram of the MPL ground operations system is shown in Figure 1.8. Downlink data from mars was processed and put in databases, e.g., the WITS database. Sequence generation began by using a sequencing tool called APGEN which generated daily high level sequences for all the lander instruments. The sequences for the different instruments were sent to sequence generation systems specific to each instrument. Included in the sequences were requests which included text descriptions of what needed to be done in each request and how much time, energy, and data volume was allocated for each request. WITS was the sequence generation system for the RA and RAC and some SSI operations. WITS generated the low-level commands to achieve the goals specified in the requests and within the resource allocations specified. The multiple sequencing systems then output their sequences to the SEQGEN planning tool where all the low-level sequences were integrated and resource checking on the integrated sequence was done and final sequence modifications were made to ensure a valid sequence within resource constraints. The final sequence was then sent into the uplink process, eventually to be received at the lander.

An important motivation for the development and use of WITS in a planetary mission is its use in public outreach. A separate version of WITS was made available to the general public to download and run on their home computers. A subset of mission data was placed in the public WITS database. Since the actual mission commands and flight software are proprietary, new arm kinematics and new commands were used in the public version. The public is then able to view mission data and plan and simulate their own missions. The site to download the public MPL mission WITS can be found at URL http://robotics.jpl.nasa.gov/tasks/wits/. The public outreach site was hosted at Graham Technology Solutions, Inc. The KEWi Internet help desk system from 3Si Inc. [16] was used to provide technical support to the public (it is also used for FIDO rover WITS technical support). With KEWi, public users can query for information about WITS using the KEWi search engine or submit service tickets which the KEWi WITS support staff then reply to.

### 1.3.2 FIDO Rover Field Test Results

A science and engineering team used WITS locally in the field to command the FIDO rover in the April 1999 field test at Silver Lake, California. Geographically distributed high school students commanded the FIDO rover during the last two days of the field test using WITS via the Internet (connected to the field via a satellite connection). High schools from Los Angeles, CA, Phoenix, AZ, Ithica, NY, and St. Louis, MO made up the LAPIS student tests mission team. The students viewed the downlink data and generated the command sequences. After the students finished generating a sequence, they submitted it to the common database and the field located operators reviewed it and sent it for execution on the rover. During field test operations, WITS had the functionality to command eight different instruments (pancam, navcam, hazcams, bellycam, IPS, color microscopic imager, moessbauer, and mini-corer), as well as the mast, arm, and rover motions. Also, WITS had the functionality to display the downlink data results for all instruments in graphical and/or textural format.

## 1.4 Conclusions and Future Work

WITS provides a new Internet-based operations paradigm for planetary mission operations. WITS was used in the Mars Polar Lander mission ground operations for downlink data visualization and command sequence generation for the robotic arm and robotic arm camera and as a secondary tool for the stereo surface imager. With the use of WITS, the MPL mission was the first planetary mission to utilize Internet-based ground operations. WITS can also provide Internet-based operations for future mars rover missions, as demonstrated in its use in the FIDO rover field test. The integrated visualization, sequence planning, and Internet security features of WITS make it a valuable tool for planetary mission operations. Also, WITS is an engaging public outreach tool which the public can use to visualize mission downlink data and plan and visualize their own missions.

WITS is currently being prepared for the next FIDO rover field test in May 2000. The primary focus of current development is to enhance the visualization and simulation capabilities, e.g., simulation of the rover traverse and simulation of rover instrument arm and mini-corer deployment against targets.

## 1.5 Acknowledgments

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The contribution by IA Tech, Inc. was supported by a contract from the NASA Small Business Innovation Research program.

## 1.6 References

[1] P. Backes, G. Tharp, and K. Tso: The Web Interface for Telescience (WITS), IEEE International Conference on Robotics and Automation, pages 411-417, Albuquerque, New Mexico, April 1997

[2] P. Backes, K. Tso, and G. Tharp: Mars Pathfinder Mission Internet-Based Operations Using WITS, IEEE International Conference on Robotics and Automation, pages 284-291, Leuven, Belgium, 1998

[3] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, and R. Petras: The Rocky7 Rover: A Mars Sciencecraft Prototype, IEEE International Conference on Robotics and Automation, pages 2458-2464, Albuquerque, New Mexico, April 1997

[4] R. Volpe: Navigation Results From Desert Field Tests of the Rocky7 Mars Rover Prototype, International Journal of Robotics Research, Special Issue on Field and Service Robots, 18(7), July 1999

[5] E. Paulos and J. Canny: Delivering Real Reality to the World Wide Web Via Telerobotics, IEEE International Conference on Robotics and Automation, pages 1694-1699, Minneapolis, Minnesota, April 1996

[6] K. Goldberg, M Mascha, S. Genter, N. Rosenberg, C. Sutter, and J. Wiegley: Robot Teleoperation Via WWW, IEEE International Conference on Robotics and Automation, May 1995

[7] NASA/UCLA: Mars Surveyor 98 Lander, http://mars.jpl.nasa.gov/msp98/lander/, 1999

[8] P. Schenker et al.: New Planetary Rovers for Long Range Mars Science and Sample Return, Intelligent Robotics and Computer Vision XVII, SPIE Proc. 3522, November, 1998.

[9] R. Arvidson et al.: FIDO: Field-Test Rover for 2003 and 2005 Mars Sample Return Missions, 30th Lunar and Planetary Science Conference, Houston, Texas, March 15-19, 1999

[10] R. Rivest, A. Shamir, and L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, 21(2), pages 120-126, February 1978

[11] SSL 3.0 Spec. http://home.netscape.com/eng/ssl3/, Netscape Corporation

[12] RSA Labs FAQ, Section 3.2: DES, http://www.rsasecurity.com/rsalabs/faq/3-2.html, RSA Security Incorporated, 1999

[13] M. Wiener: Performance Comparison of Public-Key Cryptosystems, CryptoBytes, 3(3), pages 1-5, 1998

[14] Entrust/Toolkit Java Edition, http://developer.entrust.com/java/, Entrust Technologies, 1999

[15] W. Diffie and M. Hellman: New Directions in Cryptography, IEEE Transactions on Information Theory, IT-22, pages 644-654, 1976

[16] 3Si, Inc.: http://www.3si.com

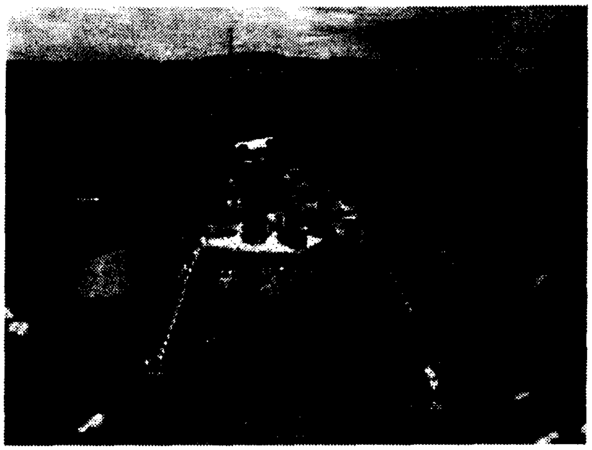

*Figure 1.1: Artist's drawing of Mars Polar Lander on mars*

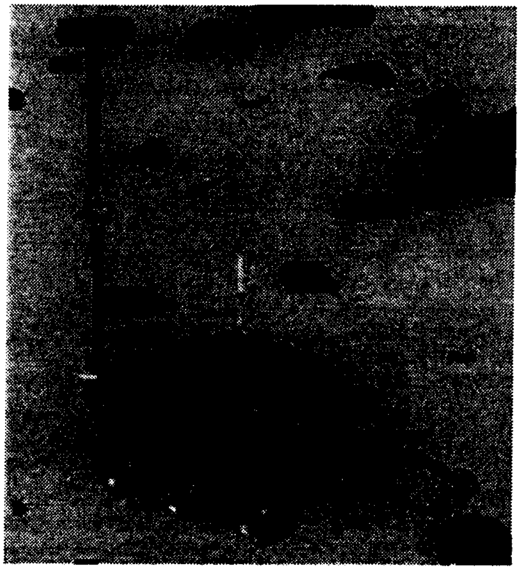

*Figure 1.2: FIDO rover*

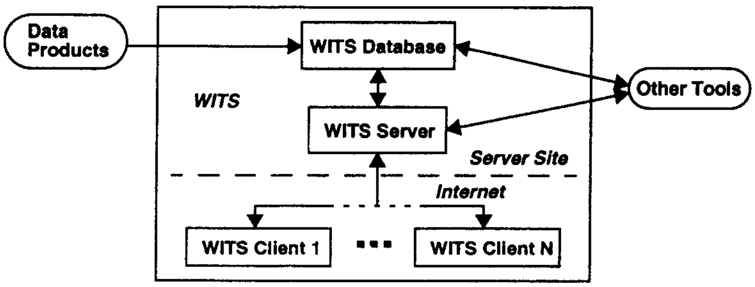


*Figure 1.3: WITS architecture*

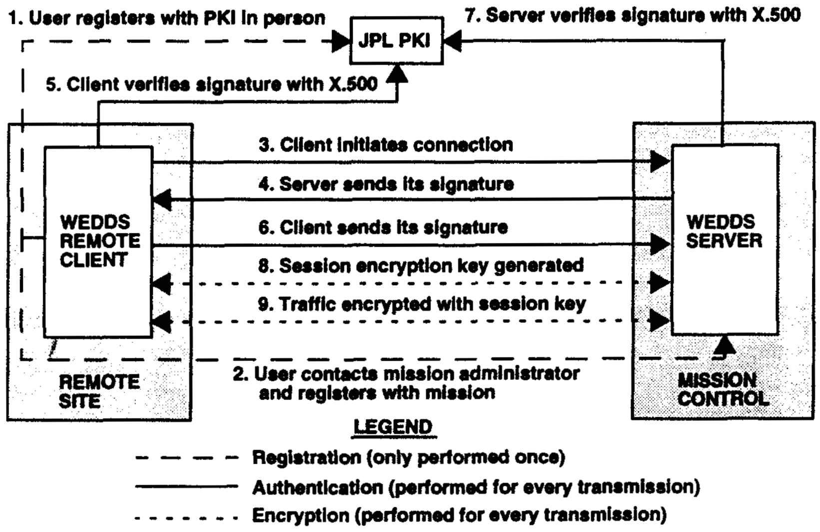


*Figure 1.4: Secure Internet communication steps*

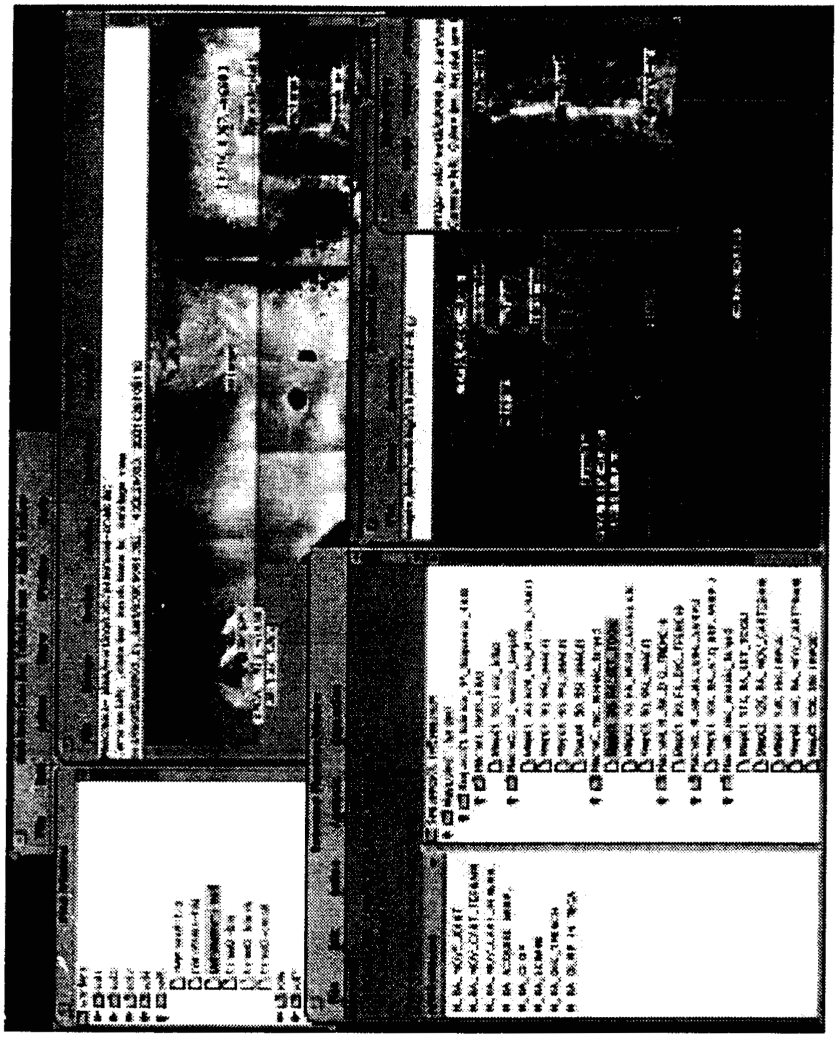

*Figure 1.5:* *Panorama, Wedge, and Overhead views and Sequence and Plan windows*


*Figure 1.6: 3D view with lander and terrain visualization*

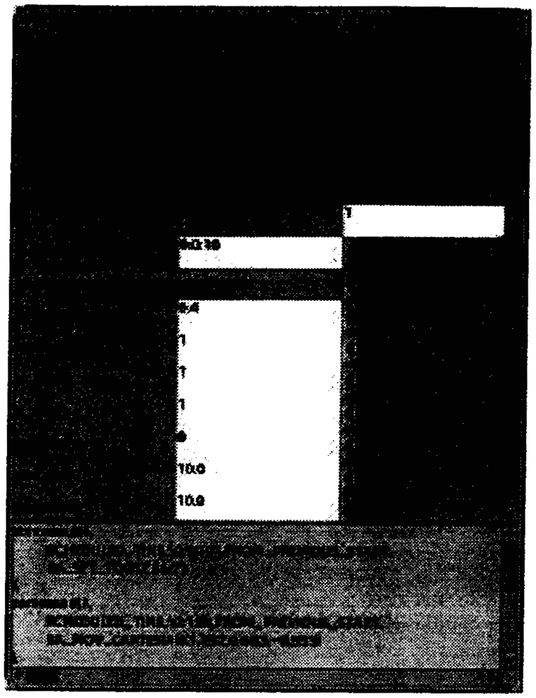

*Figure 1.7:* *Macro window*

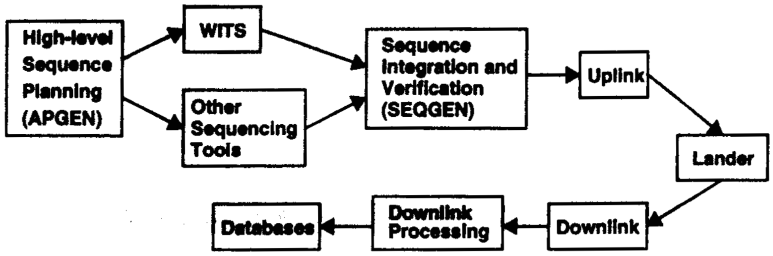


*Figure 1.8:* MPL mission operations architecture